

BY EXPRESS MAIL NO. EL387309184US
Attorney Docket No. KOIK-P9784

MENU

SEARCH

INDEX

1/1



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06282443

(43)Date of publication of application: 07.10.1994

(51)Int.Cl.

G06F 9/45
G06F 9/06
G06F 9/06

(21)Application number: 05068248

(71)Applicant:

SONY CORP

(22)Date of filing: 26.03.1993

(72)Inventor:

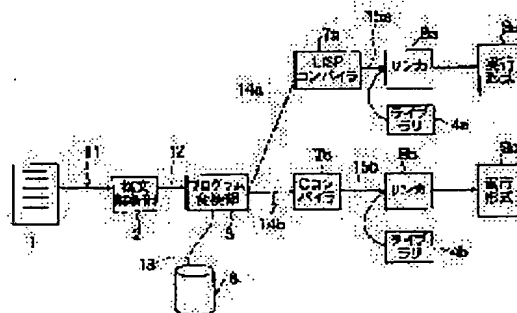
KITATANI YOSHIMICHI

(54) METHOD AND DEVICE FOR EDITING PROGRAM

(57)Abstract:

PURPOSE: To provide a compilation device and its method which enables a different language to be executed on a different processor element PE processing system with optimum processing efficiency.

CONSTITUTION: A syntax analytic part 2 analyzes the syntax of a source program 1 described in the LISP language. A program conversion part 3 generates a program 14a suitable for LISP compilation and a program 14b suitable for C compilation from the syntax analyzed program 11 by referring to the contents of a data base 6. A LISP compiler 7a and a C compiler 7b compile the inputted programs according to their language systems. A linker 8a converts the LISP-compiled program into machine word codes 9a which can be executed on a 1st program processing system by referring to a library 4a. A linker 8b, on the other hand, the C language compiled program into machine word codes 9b which can be executed on a 2nd program processing system by referring a library 4b.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of
rejection]
[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

[MENU](#)

[SEARCH](#)

[INDEX](#)

(19)日本国特許庁(J P)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-282443

(43)公開日 平成6年(1994)10月7日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/45				
9/06	4 1 0 H	9367-5B		
	4 3 0 E	9367-5B		
		9292-5B		
			G 0 6 F 9/ 44	3 2 2 A
審査請求 未請求 請求項の数 9 O L (全 10 頁)				

(21)出願番号 特願平5-68248

(22)出願日 平成5年(1993)3月26日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 北谷 義道

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

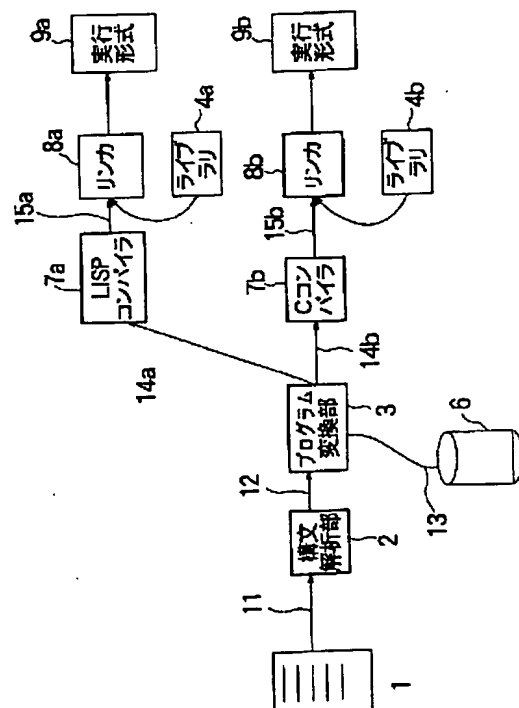
(74)代理人 弁理士 佐藤 隆久

(54)【発明の名称】 プログラム編集方法と装置

(57)【要約】

【目的】 異なる言語をその言語の処理効率を異なるプロセッサエレメントPE処理系で最適に実行することを可能ならしめるコンパイル装置とその方法を提供する。

【構成】 構文解析部2はLISP言語で記述されたソースプログラム1の構文を解析する。プログラム変換部3は構文解析されたプログラム1をデータベース6の内容を参照して、LISPコンパイラに適したプログラム14aとCコンパイラに適したプログラム14bを生成する。LISPコンパイラ7aおよびCコンパイラ7bはそれぞれ入力されたプログラムをその言語体系にしたがってコンパイルする。リンカ8aはLISPコンパイルされたプログラムをライブラリ4aを参照して第1のプログラム処理系で実行可能な機械語コード9aに変換する。リンカ8bはC言語コンパイルされたプログラムをライブラリ4bを参照して第2のプログラム処理系で実行可能な機械語コード9bに変換する。



【特許請求の範囲】

【請求項1】高級プログラミング言語で記述されたソースプログラムを、構成部分ごとに処理内容进行分析し、それらを複数のプログラム処理系に振り分けそれぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うプログラム編集方法において、

複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行なった結果を記録しているデータベースを用い、

ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索し、

それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換することを特徴とするプログラム編集方法。

【請求項2】ソースプログラム中に定義された一つの関数ブロックを、その中に含まれるプログラム処理に応じて複数のプログラム処理系のどれかに分類し、該分類されたプログラム処理系に対応するデータベースに登録し、

それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼び出しのプログラムに変換することを特徴とする請求項1記載のプログラム編集方法。

【請求項3】プログラム処理系が持つ機能に対してその実行に必要とされるプログラム処理系の資源の多寡を表現する数的データをデータベースに保持し、

そのデータを元に変換の対象となるソースプログラム中の関数呼びだしのシーケンスの実行に要するプログラム処理系の資源を計算し、

計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類し前記データベースに登録することを特徴とする請求項1または2記載のプログラム編集方法。

【請求項4】前記高級プログラミング言語で記述されたソースプログラムは、LISP言語で記述されている請求項3記載のプログラム編集方法。

【請求項5】高級プログラミング言語で記述されたソースプログラムを、構成部分ごとに処理内容进行分析し、それらを複数のプログラム処理系に振り分けそれぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うプログラム編集装置において、

複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行なった結果を記録しているデータベースを有し、

ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索する手段を有し、

それら機能ブロックの呼び出しを、対応するプログラム

処理系への機能呼び出しのプログラムに変換する手段を有する、ことを特徴とするプログラム編集装置。

【請求項6】ソースプログラム中に定義された一つの関数ブロックを、その中に含まれるプログラム処理に応じて複数のプログラム処理系のどれかに分類する手段を有し、

分類されたプログラム処理系に対応するデータベースに登録する手段を有し、

それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼び出しのプログラムに変換する手段を有することを特徴とする請求項5記載のプログラム編集装置。

【請求項7】プログラム処理系が持つ機能に対してその実行に必要とされるプログラム処理系の資源の多寡を表現する数的データをデータベースに保持し、

そのデータを元に変換の対象となるソースプログラム中の関数呼びだしのシーケンスの実行に要するプログラム処理系の資源を計算する手段を有し、

計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類し前記データベースに登録することを特徴とする請求項5または6記載のプログラム編集装置。

【請求項8】前記高級プログラミング言語で記述されたソースプログラムは、LISP言語で記述されている請求項7記載のプログラム編集装置。

【請求項9】高級プログラミング言語で記述されたソースプログラムを構成部分ごとに処理内容进行分析する構文解析部、

複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベース、

ソースプログラム中で名前によって呼び出す機能を前記データベースから検索し、それらを複数のプログラム処理系に振り分けるプログラム変換部、

それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換する編集手段を有し、それぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うことを特徴とするプログラム編集装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はデジタル計算機におけるコンパイラに関するものであり、特に複数のプログラム言語が同一の計算機システムで動作する場合におけるコンパイラおよびプログラム編集方法とその装置に関する。

【0002】

【従来の技術】従来からC言語等の高級プログラム言語で記述されたソースプログラムを機械語コードなどの低位言語に変換するコンパイラといったプログラム編集方

法が知られており、実際に使用されている。以下、コンパイラを使用し、高級プログラム言語で記述されたソースプログラムを下位言語に変換する、従来のプログラム編集方法について述べる。

【0003】図6は従来のコンパイラ装置でのデータの流れを示した図である。このコンパイラ装置は、構文解析部2、コード生成部23、アセンブラ処理部（アセンブラ）23、リンカ部25を有する。図6においてソースプログラム1はC言語で記述されたプログラムである。構文解析部2はC言語で記述されたソースプログラムの文法チェックなどを行ない、コンパイラの内部表現に変換する。次に、コード生成部23はその内部データを構文解析部2から受け取り、アセンブラコードを生成する。最後に、アセンブラ24はコード生成部23が生成したアセンブラコードを受け取ってコンピュータなどのプログラム処理系において実行可能なファイルを生成する。

【0004】次に、上記コンパイラ装置によって同じソースプログラム1内や別のソースプログラムをコンパイルすることによって得られた機能ブロックを呼び出すプログラムを処理する場合を考える。このような機能ブロックはソースプログラム中で関数と呼ばれる名前として参照され、呼び出される。構文解析部2は関数呼出しのコードを受け取ると、関数呼出し表す特別なデータ型を生成し、コード生成部23に渡す。コード生成部23は関数名に一意に対応するアセンブラコードのラベルを作るルーチンを持つ。このルーチンが関数名に対応するラベルを生成し、そのラベルへのジャンプを表すアセンブラ命令を生成する。

【0005】さらに、ソースプログラム1内の部分で、関数をブロックを定義している部分を処理する場合を考える。コード生成部23はその関数機能を表すアセンブラコードを生成する。また先に述べた、関数名に対してアセンブラコードのラベルを生成するルーチンを使ってこの機能ブロックの関数名に対応するラベルを作り、生成したアセンブラコードに付加する。

【0006】このように作成された機能ブロックの呼出しに対応するアセンブラコードに含まれるラベルの参照は、アセンブラ24によって機械語コードに翻訳されたあとリンカ25によってライブラリ26を参照して実際の番地に変換され、適切な機能ブロックの呼出しコードが生成される。このようにして生成された実行可能な形式のプログラムがプログラム処理系、具体的にはコンピュータにおいて実行される。

【0007】

【発明が解決しようとする課題】関数の呼出しコードを生成するために、関数名とラベル名だけの情報を持っている上記のコンパイラ装置による上記コンパイル方法では、ソースプログラム中に現れるそれぞれの関数や機能に対応して別々の処理系の機能と呼び出すことは想定さ

れていない。高級プログラミング言語はそれぞれ用途に向き、不向きがあり、例えば、FORTRANは数値計算に適しているが、計算機のハードウェアに密着した入出力を取り扱うといった用途には向いていない。このため、例えば、FORTRANを使用して数値演算とI/O操作の混在したプログラムを作成すると、数値計算処理は高速処理可能であるが、I/O操作処理は遅いオブジェクトプログラムが生成されるという問題が生じる。

【0008】したがって、本発明は、ソースプログラムの記述言語に応じた効率のよいオブジェクトプログラムを編集する装置とその方法を提供する。

【0009】

【課題を解決するための手段、および、作用】上記の問題を解決するために、本発明に係るコンパイラ方法は、それが想定する複数のプログラム処理系、具体的には、コンピュータ、それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベースを用い、ソースプログラム中で名前によって呼び出す機能をそのデータベースから検索し、それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼出しのプログラムに変換することを特徴とする。

【0010】また、本発明は、ソースプログラム中に定義された一つの関数ブロックを、その中に含まれる処理に応じて複数のプログラム処理系のどれかに分類し、分類されたプログラム処理系に対応するデータベースに登録し、それらの関数の呼び出しを対応するプログラム処理系中の対応する機能呼び出しのプログラムに変換することを特徴とする。

【0011】さらに、本発明は、プログラム処理系が持つ機能に対してその実行に必要なとされるプログラム処理系の資源の多寡を表現する数値データをデータベースに保持し、そのデータを元に変換の対象となるソースプログラム中の関数呼び出しのシーケンスの実行に要するプログラム処理系の資源を計算し、計算された結果を利用してプログラムの実行に必要な資源が小さくなるよう関数ブロックを適切に分類しデータベースに登録することを特徴とする。

【0012】また本発明によれば、上述したプログラム編集方法を実施するプログラム編集装置が提供される。

【0013】さらに本発明によれば、高級プログラミング言語で記述されたソースプログラムを構成部分ごとに処理内容を分析する構文解析部、複数のプログラム処理系それぞれに対し、そのプログラム処理系が持つ機能を一連の意味のある機能ブロック毎に名前付を行ない記録するデータベース、ソースプログラム中で名前によって呼び出す機能を前記データベースから検索し、それらを複数のプログラム処理系に振り分けるプログラム変換部、それら機能ブロックの呼び出しを、対応するプログラム処理系への機能呼び出しのプログラムに変換する編

集手段を有し、それぞれのプログラム処理系の機械語コードや実行可能なプログラムに変更するプログラム編集を行うことを特徴とするプログラム編集装置が提供される。

【0014】

【実施例】図面を参照して本発明の実施例について説明する。図1は、本発明のプログラム編集方法が適用されるプロセッサエレメントPE（コンピュータシステム）の概略構成図である。図1（A）は単独のプロセッサエレメントPEの構成を示す図であり、図1（B）は複数のプロセッサエレメントPEが相互に接続されて大規模コンピュータシステムを構成する図を示す。

【0015】図1（A）に示したプロセッサエレメントPEは、LISPなどの記号言語を処理する記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機RMと主記憶MMおよび二次記憶装置とを有するストーリージ（記憶装置）から構成されている。記号処理用計算機エバリュエータEUに対して、入出力や記憶管理に使われる計算機RMおよびストーリージが一種のリソース（資源）として扱われている。

【0016】図1（B）は、各プロセッサエレメントPE内の入出力や記憶管理に使われる計算機RMが、LAN（Local Area Network）などの相互接続系（インターリソースコネクション）を介して、相互に接続されて、複数のプロセッサエレメントPEが全体として、大規模コンピュータシステムを構成している。

【0017】図2は図1（A）に示されたプロセッサエレメント（PE）の構成の詳細を示す図である。本発明のプログラム編集方法は、図2に示すような記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機（RM）からなる並列計算機ネットワークのプロセッサエレメントPEに適用される。

【0018】記号処理用計算機エバリュエータEUは、データ幅32ビットのVLIW（Very Long Instruction Word）制御のプロセッサであり、ユーザ定義の関数やメソッドのコンパイルされたコードを高速実行するためのインストラクションキャッシュ（64ビット×64Kワード）を持つ。また、記号処理用計算機エバリュエータEUは、データのフィールド演算を高速に実現するためのマスカ（Masker）／算術論理処理ユニット

（ALU）を持ち、関数やメソッドの呼び出しを高速化するためのデータフィールド値による多方向分岐機能を持ち、さらに多方向分岐のためのルックアップテーブルと記号処理用計算機エバリュエータEUのシステムスタックのための64Kワードの制御用メモリを持つ。さらに記号処理用計算機エバリュエータEU、記号処理計算を実行する際のフレームスタックとレジスタファイルのための64Kワードのローカルメモリを持つ。記号処理用計算機エバリュエータEUは、入出力や記憶管理に要する演算は、入出力や記憶管理に使われる計算機RMに

その実行を要求する。

【0019】入出力や記憶管理に使われる計算機RMは、記憶管理、デバイス管理、プロセス管理、ファイル管理、入出力、他のプロセッサエレメントPEとの通信の機能を記号処理用計算機エバリュエータEUに提供する。入出力や記憶管理に使われる計算機RMには、ワークステーションWSを使用して、ワークステーションWSのXウィンドウ・システム上にアプリケーション開発時および実行時のインタラクティブなヒューマンインタフェース環境を実現する。そのため、入出力や記憶管理に使われる計算機RMは関数定義やメソッド定義のインクリメンタルな修正・コンパイル、エラー状態からの復帰・実行再開機能を持つ。さらに入出力や記憶管理に使われる計算機RMは、アプリケーションの実行時にユーザとの動的な対話ができるような入出力機能を提供する。入出力や記憶管理に使われる計算機RMは、主記憶MM上の記号処理データの高速アクセスのためのポイント・マニピレータ（Pointer Manipulator）を持つ。入出力や記憶管理に使われる計算機RMは、並列処理時に他のプロセッサエレメントPEとの通信を行なうための通信インタフェースを装着可能である。

【0020】記憶装置（ストーリージ）のうち主記憶MMは、記号処理用計算機エバリュエータEUからも直接アクセス可能なデュアルポートメモリである。主記憶MMでは、排他的アクセス、記号処理演算で有効な参照カウントなどの機能アクセスを提供する。主記憶MMは、これらの制御情報と2ワードのポインタを表現するため、80ビット×256Kワードの構成となっている。

【0021】記憶装置のうち二次記憶は、ワークステーションWSのディスクシステムを使用する。

【0022】記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機RMとは、入出力や記憶管理に使われる計算機インターフェースRM-I/Fで接続され、記号処理用計算機エバリュエータEUから入出力や記憶管理に使われる計算機RMに、また入出力や記憶管理に使われる計算機RMから記号処理用計算機エバリュエータEUに機能データを送出する。また、記号処理用計算機エバリュエータEUからメインメモリインターフェースMM-I/Fを介して情報データが送出され、主記憶MMと入出力や記憶管理に使われる計算機RMに送出される。

【0023】入出力や記憶管理に使われる計算機インターフェースRM-I/F、および、主記憶インターフェースMM-I/Fは、記号処理用計算機エバリュエータEU内の内部バスIBUSを介して算術論理処理ユニットALUに接続される。また、入出力や記憶管理に使われる計算機インターフェースRM-I/Fは、入出力や記憶管理に使われる計算機RM内のエバリュエータインターフェースEU-I/Fを介してワークステーションWSに接続される。

【0024】図3は本発明のコンパイラ装置のプログラム変換形態を表す図である。このコンパイラ装置は、構文解析部2、プログラム変換部3、LISPコンパイル処理部（LISPコンパイラ）7a、Cコンパイル処理部（Cコンパイラ）7b、LISP言語に関するリンケージ処理を行う第1のリンカ8a、および、C言語に関するリンケージ処理を行う第2のリンカ8bを有する。

【0025】ソースプログラム1はLISPで記述されたソースプログラムである。構文解析部2はLISPで記述されたソースプログラムの文法チェックなどを行ない、構文木データなどのコンパイラの内部表現に変換するモジュールである。プログラム変換部3はその内部データを構文解析部2から受け取り、図2に示した記号処理用計算機エバリュエータEUおよび入出力や記憶管理に使われる計算機RMで実行されるべきプログラムコードを生成するモジュールである。LISPコンパイラ7aとCコンパイラ7bはそれぞれ、ライブラリ6a、6bを参照してプログラム変換部3が生成したプログラムコードを、リンカ8a、8bを介して、記号処理用計算機エバリュエータEUおよび入出力や記憶管理に使われる計算機RMなどのプログラム処理系、つまり、コンピュータで実行可能な機械語コード（オブジェクトプログラム）9a、9bに変換するコンパイラである。機械語プログラム9a、9bはコンパイラ7a、7bが生成した機械語プログラムであり、それぞれ図2の記号処理用計算機エバリュエータEUおよび入出力や記憶管理に使われる計算機RMに読み込まれて実行される。データベース6は、図2に示した記号処理用計算機エバリュエータEUおよび入出力や記憶管理に使われる計算機RMで実行される関数の情報を蓄積するデータベースであり、プログラム変換部3が関数名をキーとしてその関数が記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機RMのどちらの処理系で実行されるかを検索するために使われる。

【0026】以下、本発明のコンパイラ装置の動作について説明する。図4および図5は本発明のコンパイラ装置を用いてプログラム編集する編集方法の処理内容を示すフローチャートである。図4および図5はそれぞれ図3に示したプログラム変換部3におけるコード生成の処理を表し、図4の処理によってLISP言語のコードが生成され、図5のフローチャートに示した処理によってC言語のコードが生成される。

【0027】図4は、図3の構文解析部2からプログラムの解析結果12を受け取りLISPコードを生成する処理を表す。また、プログラム変換部3は解析結果12が無くなるまで図4の処理を繰り返し実行する。プログラム変換部3は、ステップ01（S01）においてプログラムが関数定義を表すかどうかを判断する。関数定義でなければ、プログラム変換部3は、ステップ02（S02）において式が関数呼出しかどうかを判断する。そ

うでなければ、プログラム変換部3は、S03において通常のLISP処理としてそのまのLISPコードを生成し出力する。この出力は図3に示したLISPコンパイラ7aに渡される。S01において対象プログラムが関数定義であると判断すると、構文解析部2は、S05でそのプログラムを解析する。そしてS06において、プログラム変換部3は、そのプログラムが図2の記号処理用計算機エバリュエータEUで実行されるLISPコード14aとして出力するか、あるいは図2の入出力や記憶管理に使われる計算機RMで実行されるCコード14bとして出力するかを決定する。LISPコード14aとして出力する場合、プログラム変換部3は、S07の処理でそのプログラムはLISPコード14aとしてLISPコンパイラ7aに対して出力する。Cコード14bとして出力する場合、プログラム変換部3は、S09で図5のCコード生成処理を呼出しCコード14bに変換する。

【0028】S02においてプログラム変換部3が関数呼出し式を認識すると、プログラム変換部3は、S10において呼びだされる関数が記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機RMのどちらの処理系に属しているかを判断する。このとき、プログラム変換部3は、図3のデータベース6を検索する。もし入出力や記憶管理に使われる計算機RMに属している関数であれば、プログラム変換部3は、S11においてRM関数を呼び出すためのLISPコード14aを生成する。もし記号処理用計算機エバリュエータEUに属している関数であれば、プログラム変換部3は、S12においてLISP関数呼出しの形のままLISPコード14aとして出力する。

【0029】図3のプログラム12として渡される構文解析プログラムは多くの部分プログラムから構成されていることがあるので、S03とS07でそれに対するコードを生成する場合に、部分プログラムすべてに対して同様に図4の処理を再帰的に行なう。

【0030】図5の処理内容は、図4を参照して述べたLISPコード14aを生成する処理と同様であり、LISPコード14aを生成する処理を示した図4の処理と同様の処理をCコード14bを生成するために行ない、C言語コード14bを出力する。

【0031】以上の動作によって得られた次段階プログラムはそれぞれLISP用コンパイラ7a、C用コンパイラ7b、および、リンカ8aおよびリンカ8bによって機械語プログラム9a～bに変換される。以上によって、図3のソースプログラム1に対してその機能によって分類された記号処理用計算機エバリュエータEU、入出力や記憶管理に使われる計算機RMそれぞれで実行される機械語プログラム9a、9bが得られる。これらは、このプログラムにより記号処理用計算機エバリュエータEUおよび入出力や記憶管理に使われる計算機RM

は交互にお互いの関数を呼び合い処理を実行する。

【0032】上述した実施例は、図1(A)に示したコンピュータシステム、つまり、記号処理用計算機エバリュエータEUと入出力や記憶管理に使われる計算機RMとによって構成される単体のプロセッサエレメントPEにおける処理を示したが、本発明の実施に際しては、図1(B)に図解した複数のプロセッサエレメントPEで構成される大規模なコンピュータシステムにおいて、あるプロセッサエレメントPEの記号処理用計算機エバリュエータEUと他のプロセッサエレメントPEの入出力や記憶管理に使われる計算機RMとで上述のように生成された機械語コードを実行させることもできる。

【0033】

【発明の効果】以上述べたように本発明のプログラム編集装置およびその方法によれば、一種類の高級プログラミング言語で記述されたソースプログラムについて、ソースプログラムの部分毎の処理内容を分析し、その処理が呼び出す機能に応じて適切な処理系を選択し、またその処理が定義する機能を適切な処理系に分類し、処理内容に応じて最適な処理系を選択することで、効率良く高速に処理を行わせることを可能にするプログラム編集装置とその方法が提供される。

【図面の簡単な説明】

【図1】本発明が適用される計算機ネットワークの接続形態を示す図であり、(A)は、プロセッサエレメントPE単体の構成図であり、(B)は複数のプロセッサエレメントPEが大規模コンピュータシステムを構成する図である。

【図2】本発明が適用されるプログラム処理系（計算機ネットワーク）の構成図である。

【図3】本発明のプログラム編集装置におけるプログラム変換方法を示す図である。

【図4】本発明のプログラム編集方法の処理の第1例としてLISPコードを生成する処理を示すフローチャートである。

【図5】本発明のプログラム編集方法の処理の第2例としてCコードを生成する処理を示すフローチャートである。

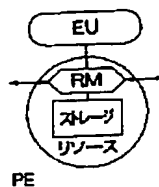
【図6】従来のプログラム編集装置の構成を表す図である。

【符号の説明】

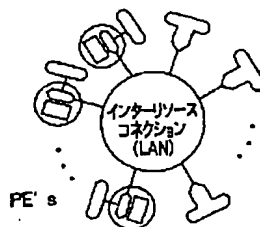
- 1・・・ソースプログラム
- 2・・・構文解析部
- 3・・・プログラム変換部
- 4a, 4b・・・ライブラリ
- 6・・・データベース
- 7a・・・LISPコンパイラ
- 7b・・・Cコンパイラ
- 8a・・・LISPリンカ
- 8b・・・Cリンカ
- 9a・・・LISPコードによる実行可能な形式のプログラム（機械語コード）
- 9b・・・Cコードによる実行可能な形式のプログラム（機械語コード）
- RM・・・入出力や記憶管理に使われる計算機
- EU・・・記号処理用計算機エバリュエータ

【図1】

(A)

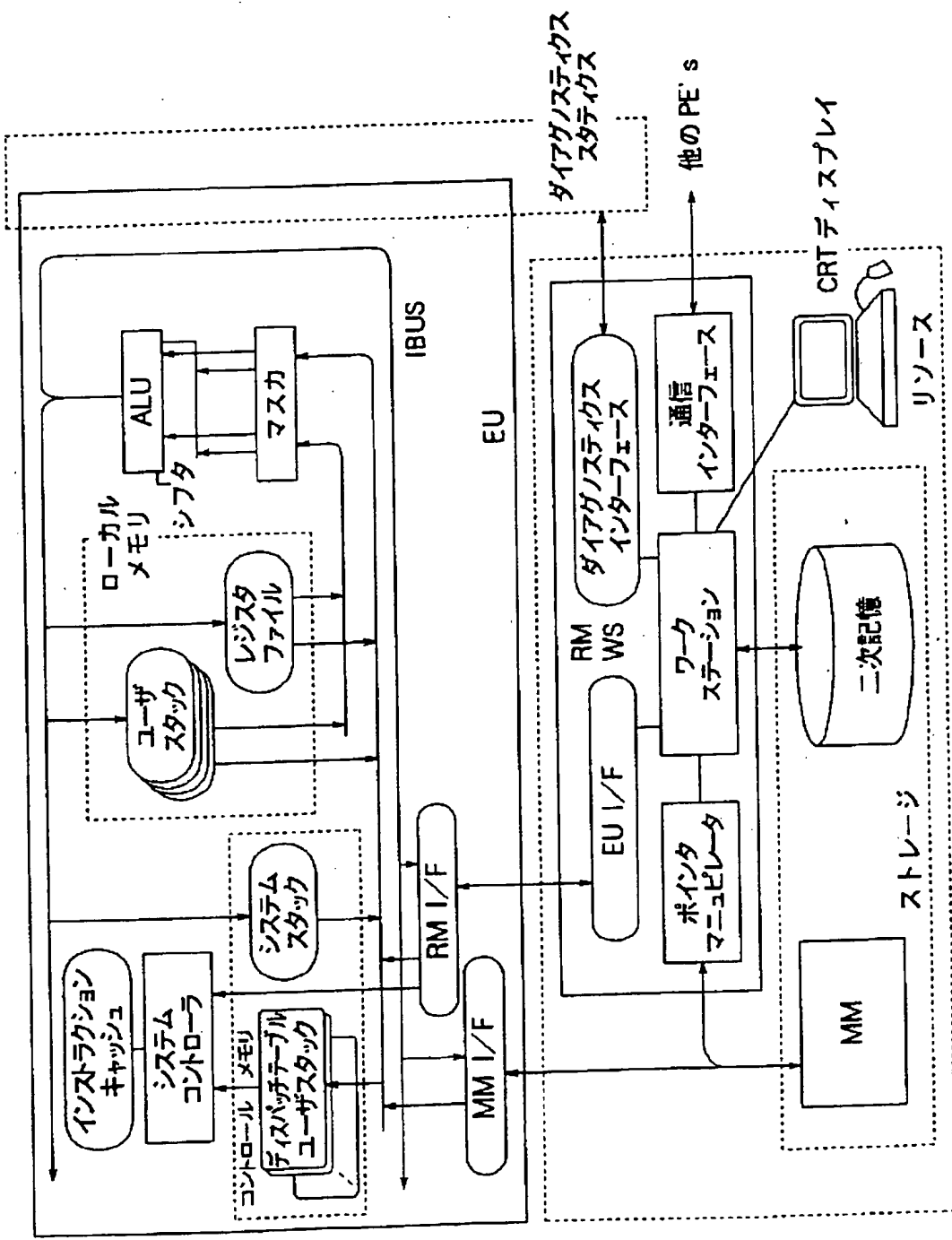


(B)

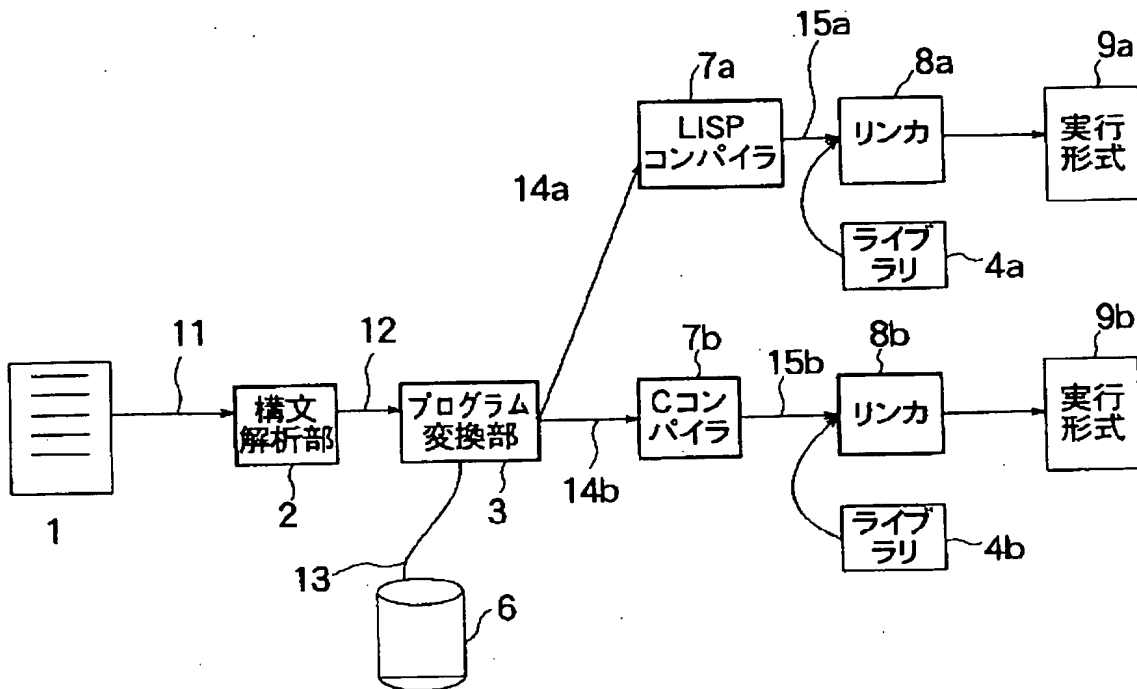


(7)

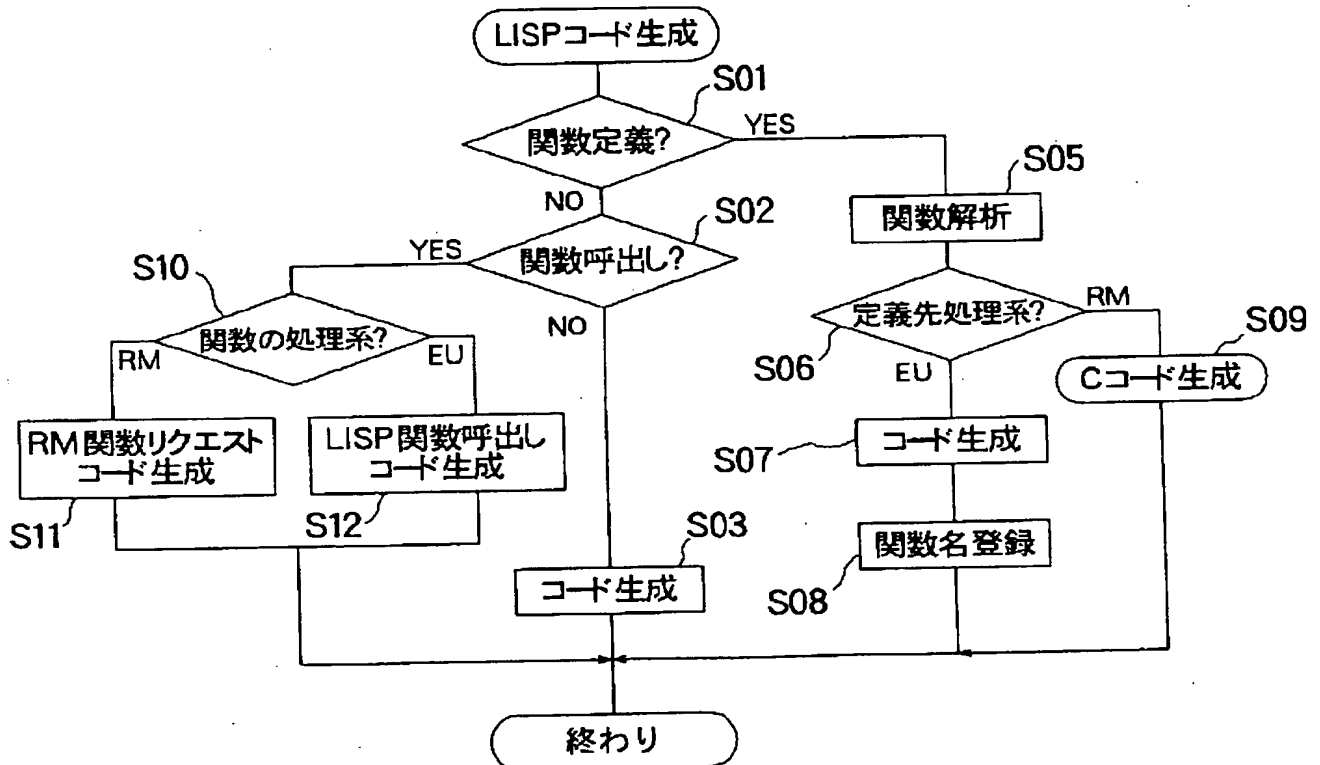
【図2】



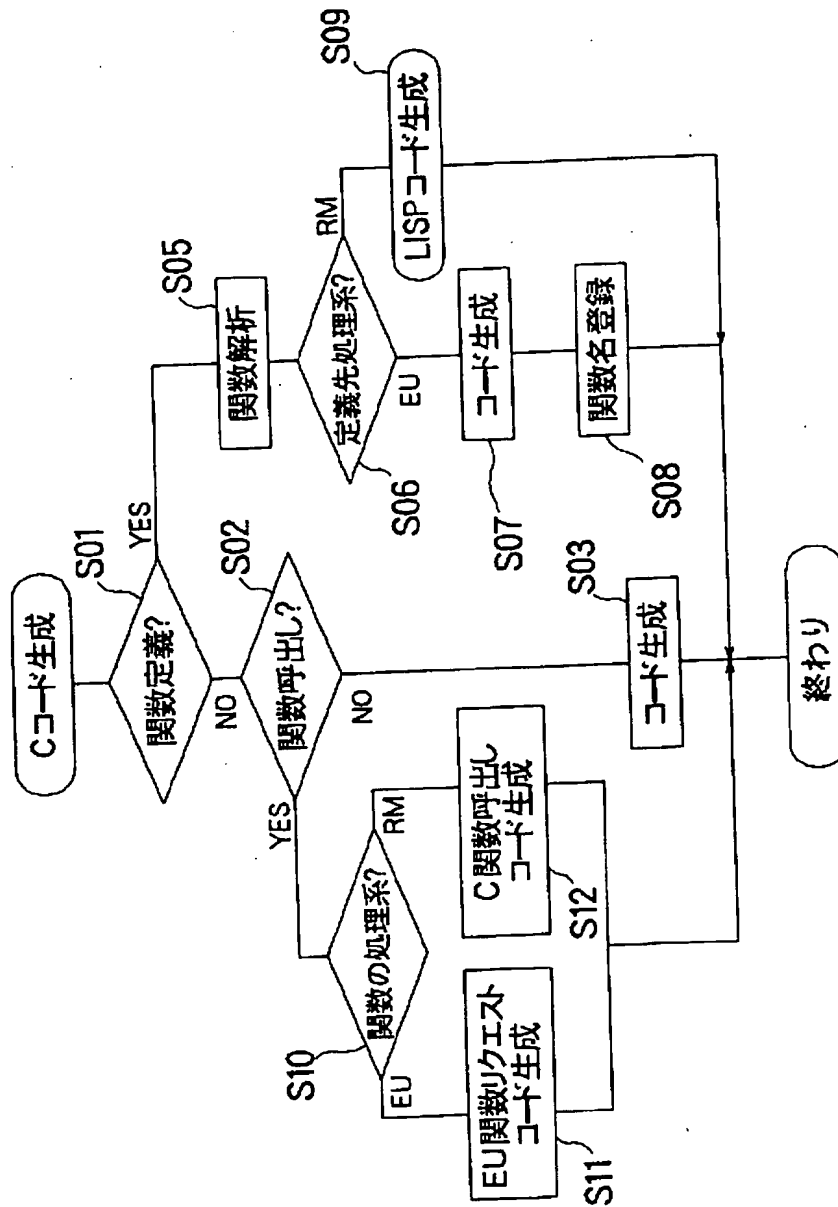
【図 3】



【図 4】



【図5】



【図6】

